The Report committee for Siddarth Bhargava Kaki certifies that this is the approved version of the following report:

On-Orbit Pose and Angular Velocity Estimation

APPROVED BY

SUPERVISING COMMITTEE:

Maruthi R. Akella, Supervisor

Renato Zanetti

On-Orbit Pose and Angular Velocity Estimation

by

Siddarth Bhargava Kaki

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2020

Acknowledgments

I wish to thank my advisor Dr. Maruthi Akella for his enduring guidance, support, and vision. I wish to also thank my second reader Dr. Renato Zanetti for his valuable suggestions and inputs.

My heartfelt gratitude goes to my family for their lifelong love and support. Many thanks to all my friends, colleagues, and fellow students for helping make every day brighter. Of course, thanks are due to all my professors and to the Department of Aerospace Engineering and Engineering Mechanics.

Finally, I offer my gratitude and efforts to my gurus and to Bhagavān.

kāyena vācā manasendriyairvā buddhyātmanā vā prakṛtessvabhāvāt |
karomi yadyat sakalaṁ parasmai nārāyaṇāyeti samarpayāmi ||
śrīman nārāyaṇāyeti samarpayāmi ||

This material is based upon work supported in part by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1610403. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

On-Orbit Pose and Angular Velocity Estimation

by

Siddarth Bhargava Kaki, M.S.E.
The University of Texas at Austin, 2020

Supervisor: Maruthi R. Akella

The problem of estimating relative pose and angular velocity for uncooperative space objects using camera images has garnered great interest, especially within applications such as asteroid mapping and satellite servicing. This report consists of two parts which address the aforementioned problem: (1) the development of a relative pose estimation and filtering pipeline for the NASA Seeker cubesat program, and (2) the theoretical development of a batch estimator based on relative orientation measurements to estimate not only the angular velocity magnitude and spin-axis direction of a target body, but also the accompanying uncertainty bounds for the resulting spin-axis direction estimate under reasonable assumptions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Pose Estimation and Filtering for the NASA Seeker Cubesat Program

1.1 Introduction

This chapter of the report describes a relative pose estimation and filtering pipeline developed for the Seeker cubesat program. Seeker is a NASA Johnson Space Center (JSC) led technology demonstration mission which aims to engage a cubesat to autonomously perform routine health and system-state inspections for the International Space Station (ISS). JSC has partnered with The University of Texas at Austin (UT) to develop vision-based navigation software for Seeker. The first iteration of the program, the Seeker-1 cubesat (Fig. 1.1), flew its mission in September 2019 [1], for which UT provided a visual machine-learning-based relative bearing estimation software package [2]. Seeker-1 deployed from and performed its navigation mission about the Cygnus Enhanced cargo resupply ship (Fig. 1.2).

Post Seeker-1, the UT team comprising of the Controls Group for Distributed and Uncertain Systems (CDUS)[1] and the Texas Spacecraft Laboratory (TSL)[2] has continued development of the vision-based navigation system,

[1] https://sites.utexas.edu/cdus/
[2] https://sites.utexas.edu/tsl/

to expand capability from providing just relative bearing to a target to also providing filtered, full relative pose estimates.

The remainder of the chapter is organized as follows. First, the architecture employed for the end-to-end pose estimation pipeline is presented in the Full Pose Estimator Architecture section. Next, the Kalman filtering components are summarized in the Full Pose Filtering section. Results from a run-time performance test and two simulated pose tracking tests are provided in the Simulated Results section. Finally, concluding remarks and notes for future work are made in the Conclusion section.

Figure 1.1: Seeker-1 Cubesat with Kenobi Relay System (Credit: NASA)

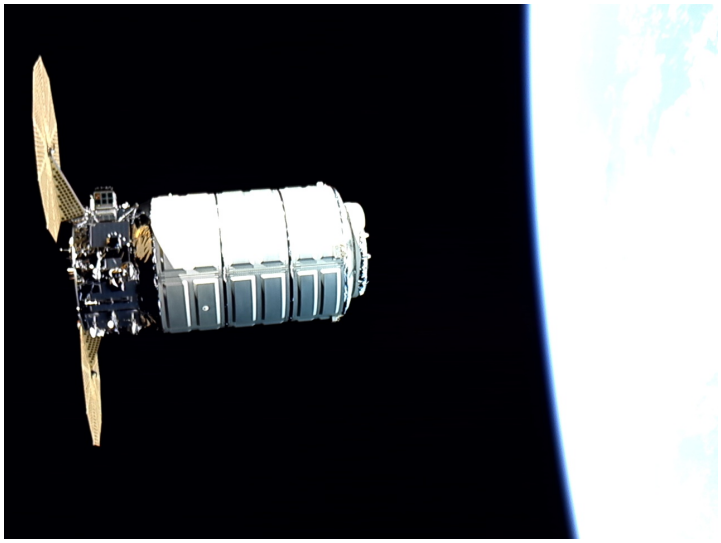


Figure 1.2: View of Cygnus Enhanced Vehicle from Seeker-1 (Credit: NASA)

1.2 Full Pose Estimator Architecture

1.2.1 Software Architecture

The high-level software architecture adopted is depicted in Fig. 1.3. A stream of images taken by a monocular camera is fed into a convolutional-neural-network (CNN), which is trained to output the relative azimuth (az.) and elevation (el.) angles to a pre-determined set of feature points on the target spacecraft body (e.g., Cygnus). These relative bearing angles (az., el.) are then fed to a pose estimation solver. Two underlying solvers are implemented: (1) a non-linear least-squares (NLS) solver, and (2) a perspective-n-point (PnP) solver. The NLS solver provides the maximum likelihood estimate of the full relative pose of the target body, while the PnP solver provides the numerical solution to the EPnP algorithm [3]. These pose estimates are then treated

as "measurements" for a Kalman filtering loop, which filters the relative pose estimates. The filtered pose estimates are then outputted, and for example, are written to disk. The software supports either: (1) locally simulating measurements for testing purposes, or (2) sourcing measurements from an external process (e.g., from a sensor), which are then provided to the main process as a serialized data stream over a named pipe. The serialization is accomplished via Protocol Buffers[1]. The software also supports basic memory management by logging filtered pose solutions to disk and subsequently clearing program memory periodically.

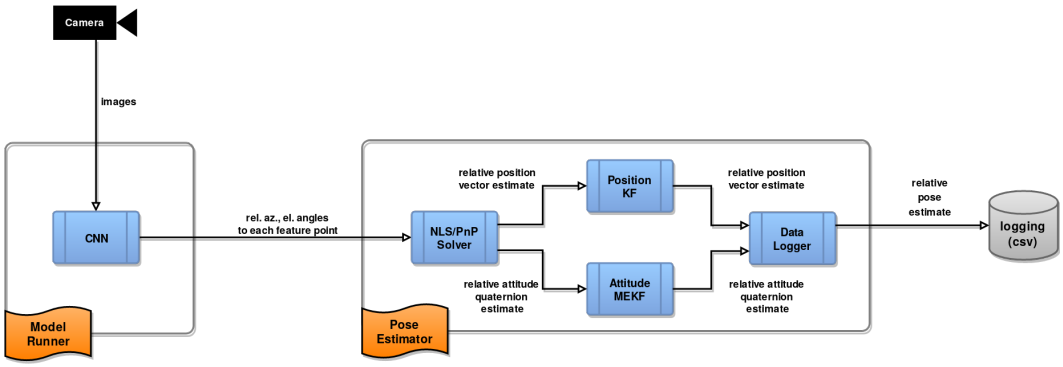


Figure 1.3: Seeker Software Architecture

The "Pose Estimator" set of components indicated in Fig. 1.3 was developed and implemented in C++ as a general-use-case open-source project called `pose_terrier`[2]. The "Model Runner" component is being developed by colleagues at TSL, and the work has been accepted for publication [4].

[1] `https://developers.google.com/protocol-buffers/`
[2] `https://github.com/siddarthkaki/pose_terrier`

1.2.2 NLS Solver

The NLS solver is implemented with the following libraries: Eigen [5] (for linear algebra), and Google's Ceres Solver [6] (for non-linear optimization). The NLS problem is formulated as the following.

The origin of the chaser (e.g., Seeker) frame is denoted O_c, where the chaser z-axis is along the camera boresight (see Fig. 1.4). The origin of the target (e.g., Cygnus) frame is denoted O_t. The true relative translation at any time from the chaser to the target in the chaser frame is denoted by the position vector $\boldsymbol{r}_{rel}^{c}$, and the true relative attitude at any time from the chaser to the target is denoted by the rotation matrix $\boldsymbol{T}_{c}^{t} = (\boldsymbol{T}_{t}^{c})^{T}$. The relative position estimate is denoted $\hat{\boldsymbol{r}}_{rel}^{c}$, and the relative attitude estimate is denoted by $\hat{\boldsymbol{T}}_{c}^{t}$. The fixed position of the camera in the chaser frame is denoted by $\boldsymbol{r}_{cam}^{c}$.

The fixed position of a 3D feature point i in the target body frame is denoted $\boldsymbol{r}_{i}^{t}$. Therefore, the true position of a 3D feature point i, with respect to the camera, expressed in the chaser frame, is given by:

$$\boldsymbol{r}_{i}^{c} = \boldsymbol{r}_{rel}^{c} - \boldsymbol{r}_{cam}^{c} + T_{t}^{c}\boldsymbol{r}_{i}^{t} \tag{1.1}$$

$$\boldsymbol{r}_{i}^{c} = \begin{bmatrix} x_{i}^{c} & y_{i}^{c} & z_{i}^{c} \end{bmatrix}^{T} \tag{1.2}$$

The estimated position of the same 3D feature point i, with respect to the camera, expressed in the chaser frame, is given by:

$$\hat{\boldsymbol{r}}_i^c = \hat{\boldsymbol{r}}_{rel}^c - \boldsymbol{r}_{cam}^c + \hat{T}_t^c \boldsymbol{r}_i^t \tag{1.3}$$

$$\hat{\boldsymbol{r}}_i^c = \begin{bmatrix} \hat{x}_i^c & \hat{y}_i^c & \hat{z}_i^c \end{bmatrix}^T \tag{1.4}$$

The NLS cost function is defined as:

$$\underset{\boldsymbol{r}_{rel}^c,\, \boldsymbol{T}_c^t}{min}\, J = \underset{\boldsymbol{r}_{rel}^c,\, \boldsymbol{T}_c^t}{min} \left(\sum_i \left[(\alpha_i - \hat{\alpha}_i)^2 + (\varepsilon_i - \hat{\varepsilon}_i)^2 \right] \right) \tag{1.5}$$

where:

$$\alpha_i = \arctan\left(\frac{x_i^c}{z_i^c}\right); \qquad \hat{\alpha}_i = \arctan\left(\frac{\hat{x}_i^c}{\hat{z}_i^c}\right) \tag{1.6}$$

$$\varepsilon_i = \arctan\left(\frac{y_i^c}{z_i^c}\right); \qquad \hat{\varepsilon}_i = \arctan\left(\frac{\hat{y}_i^c}{\hat{z}_i^c}\right) \tag{1.7}$$

The measured relative bearing angles α_i, ε_i are modeled as:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{w} \tag{1.8}$$

where the non-linear measurement model is:

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \alpha_1 & \varepsilon_1 & \dots & \alpha_n & \varepsilon_n \end{bmatrix}^T \tag{1.9}$$

where n is the number of feature points, and the additive noise $\boldsymbol{w}$ is modeled as a white Gaussian sequence with covariance matrix $R_{bearing}$.

The NLS implementation includes support for multiple random initializations in the attitude component, as set by a parameter read in at run-time. This eliminates the need for any attitude initial guess for the attitude estimate to converge. However, a "good enough" position initial guess is still needed for the position estimate to converge.

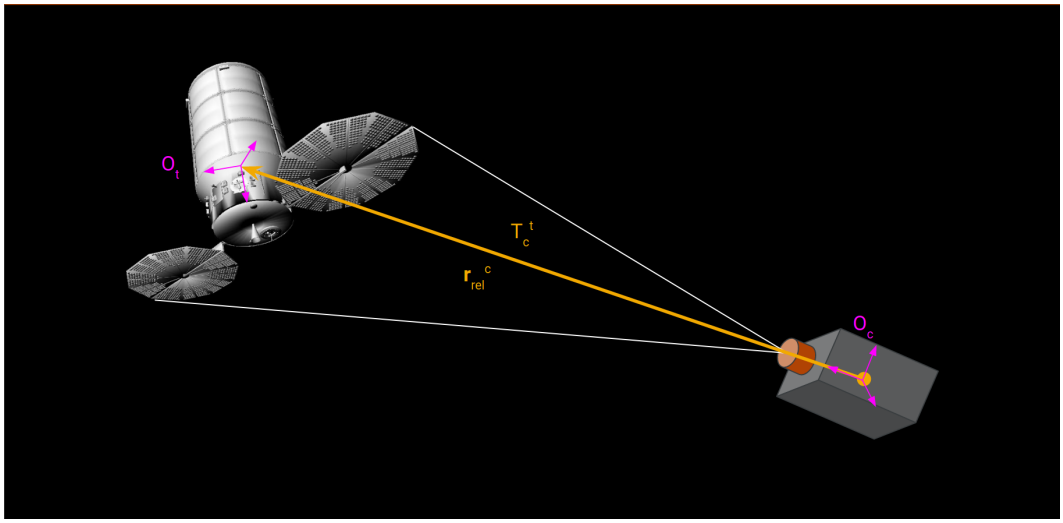


Figure 1.4: Relative Geometry

1.2.3 PnP Solver

The PnP solver is implemented with the following libraries: Eigen [5] (for linear algebra), and OpenCV [7] (for underlying PnP). The PnP problem is formulated as the following.

Recall that the fixed position of a 3D feature point i in the target body frame is denoted by $\boldsymbol{r}_i^t$. The position of the same feature point i with respect to the chaser camera, expressed in the camera frame, can be formulated in

homogeneous coordinates as:

$$\boldsymbol{r}_i^{cam} = \begin{bmatrix} T_t^c & | & \boldsymbol{r}_{rel}^c \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_i^t \\ 1 \end{bmatrix} \tag{1.10}$$

Given the 3D position of a feature point i in the camera frame $\boldsymbol{r}_i^{cam}$, and assuming a pinhole camera projection model with no lens distortion, the 2D position of feature point i in the camera image plane $(x_i^{cam},\ y_i^{cam})$ is given by:

$$\begin{bmatrix} x_i^{cam} \\ y_i^{cam} \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{r}_i^{cam} \tag{1.11}$$

up to a scale factor s, where $(f_x,\ f_y)$ describes the camera focal length in the x and y image plane directions, respectively, and $(c_x,\ c_y)$ describes the camera image plane center. Putting Eqs. 1.10 and 1.11 together leads to the direct relation between the target-frame body-fixed position $\boldsymbol{r}_i^t$ of a feature point i, and the corresponding 2D camera image plane location $(x_i^{cam},\ y_i^{cam})$:

$$\begin{bmatrix} x_i^{cam} \\ y_i^{cam} \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_t^c & | & \boldsymbol{r}_{rel}^c \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_i^t \\ 1 \end{bmatrix} \tag{1.12}$$

OpenCV's PnP solver is operated on Eq. 1.12 with Random Sample Consensus (RANSAC)[8] for robustness to solve for $\boldsymbol{r}_{rel}^c$ and $T_c^t = (T_t^c)^T$.

1.3 Full Pose Filtering

A full pose filtering loop is implemented over the static pose estimates generated by either the NLS or the PnP solver. The filtering achieves the following objectives; it provides: (1) smoothed estimates at a higher frequency than the frequency at which measurements are made, (2) a kinematic model for the underlying dynamics, and (3) a statistical method to suppress the effect of outliers on the relative pose estimates. The filtering is divided between a Linear Kalman Filter (LKF) for the position estimation, and a Multiplicative Extended Kalman Filter (MEKF) for the attitude estimation, both of which are summarized as follows.

1.3.1 Linear Kalman Filter For Position

A Linear Kalman Filter is implemented to perform relative position tracking [9]. The filter states are the following:

$$\boldsymbol{x} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T \tag{1.13}$$

where:

$$\boldsymbol{r}^c_{rel} = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{1.14}$$

$$\dot{\boldsymbol{r}}^c_{rel} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{1.15}$$

$$\ddot{\boldsymbol{r}}^c_{rel} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T \tag{1.16}$$

A discrete-time Wiener-process acceleration dynamics model is assumed with time-step Δt, which is given by:

$$\boldsymbol{x}_{k+1} = F\boldsymbol{x}_k + \boldsymbol{v}_k \tag{1.17}$$

$$F = \begin{bmatrix} F' & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & F' & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & F' \end{bmatrix}_{9\times9} \tag{1.18}$$

$$F' = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{1.19}$$

where i.i.d. noise $\boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0},\ Q)$, and process noise covariance matrix Q is given by:

$$Q = \begin{bmatrix} Q' & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & Q' & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & Q' \end{bmatrix}_{9\times9} \tag{1.20}$$

$$Q' = \sigma_v^2 \begin{bmatrix} \frac{1}{4}(\Delta t)^4 & \frac{1}{2}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{2}(\Delta t)^3 & (\Delta t)^2 & \Delta t \\ \frac{1}{2}(\Delta t)^2 & \Delta t & 1 \end{bmatrix} \tag{1.21}$$

The measurement model is given by:

$$\boldsymbol{z}_{k+1} = \boldsymbol{r}_{m,k+1} = \boldsymbol{r}_{true,k+1} + \boldsymbol{w}_{k+1} \tag{1.22}$$

$$= \begin{bmatrix} x_{m,k+1} & y_{m,k+1} & z_{m,k+1} \end{bmatrix}^T \tag{1.23}$$

where i.i.d. noise $\boldsymbol{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, R)$. Position "measurements" $\boldsymbol{r}_m = \hat{\boldsymbol{r}}^c_{rel}$ are taken as either the NLS or the PnP pose solutions. The filter propagation and measurement update steps follow the standard LKF equations.

1.3.2 Multiplicative Extended Kalman Filter for Attitude

A Multiplicative Extended Kalman Filter is implemented to perform relative attitude tracking [10, 11, 12, 13]. Only a high-level summary of the MEKF implementation is provided here; the details of the implementation follow the presentations in the references. The filter states are the following:

- Quaternion for the global attitude representation:
  $\boldsymbol{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T = \begin{bmatrix} q_w & \boldsymbol{q}_{vec}^T \end{bmatrix}^T = \boldsymbol{q}\left(T_c^t\right)$
- Gibbs vector for the error attitude representation: $\delta\boldsymbol{g} = \delta\boldsymbol{q}_{vec}/\delta q_w$
- Angular velocity vector: $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$
- Angular acceleration vector: $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z \end{bmatrix}^T$

The dynamics model employed, with time-step Δt, is given by:

$$\boldsymbol{q}_{k+1} = A_k \boldsymbol{q}_k \tag{1.24}$$

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \boldsymbol{\alpha}_k \Delta t + \boldsymbol{v}_k^{\omega} \tag{1.25}$$

$$\boldsymbol{\alpha}_{k+1} = e^{-\Delta t/\tau}\boldsymbol{\alpha}_k + \boldsymbol{v}_k^{\alpha} \tag{1.26}$$

where the attitude noise propagation is for the error attitude state (not the global), the angular acceleration dynamics (Eq. 1.26) is modeled as a discrete-time 1^{st}-order Gauss-Markov process with time constant τ, and:

$$A_k = \cos\left(\frac{1}{2}\Omega(\Delta t)\right)\mathbb{I}_{4\times 4} + \sin\left(\frac{1}{2}\Omega(\Delta t)\right)\hat{\boldsymbol{\omega}}\otimes \tag{1.27}$$

$$\hat{\boldsymbol{\omega}}\otimes = \begin{bmatrix} 0 & -\hat{\boldsymbol{\omega}}^T \\ \hat{\boldsymbol{\omega}} & -[\hat{\boldsymbol{\omega}}\times] \end{bmatrix}_{4\times 4} \tag{1.28}$$

$$\Omega = \|\hat{\boldsymbol{\omega}}\|_2 \tag{1.29}$$

where $\hat{\boldsymbol{\omega}}$ is the current estimate of the angular velocity vector. The covariance propagation is given by:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \tag{1.30}$$

$$F_k = \begin{bmatrix} -e^{(\Delta t\cdot[\hat{\boldsymbol{\omega}}\times])} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbb{I}_{3\times 3} & \Delta t\cdot\mathbb{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & e^{-\Delta t/\tau}\cdot\mathbb{I}_{3\times 3} \end{bmatrix}_{9\times 9} \tag{1.31}$$

$$Q_k = \sigma_v^2 \begin{bmatrix} \mathbb{I}_{6\times 6} & \mathbf{0}_{6\times 3} \\ \mathbf{0}_{3\times 6} & \frac{\tau\cdot Q_{psd}}{2}\cdot\left(1-e^{-2\Delta t/\tau}\right)\cdot\mathbb{I}_{3\times 3} \end{bmatrix}_{9\times 9} \tag{1.32}$$

where the skew-symmetric cross-product matrix for arbitrary vector $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ is defined by:

$$[\boldsymbol{v}\times] := \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{1.33}$$

The employed measurement model is characterized as follows. The measurement innovation is given by:

$$\boldsymbol{\nu}_{k+1} = 2\delta\boldsymbol{g}_{k+1} = 2\frac{\delta\boldsymbol{q}_{vec}}{\delta q_w} \tag{1.34}$$

where $\delta\boldsymbol{q} = \boldsymbol{q}_{m,k+1} \otimes \hat{\boldsymbol{q}}^{-1}_{k+1|k}$. The measurement sensitivity matrix is given by:

$$H_{k+1} = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbf{0}_{3\times6} \end{bmatrix}_{3\times9} \tag{1.35}$$

which is in terms of the error attitude state. The attitude "measurements" $\boldsymbol{q}_m = \boldsymbol{q}\left(\hat{T}^t_c\right)$ are taken from either the NLS or the PnP pose solutions. The filter propagation, measurement update, and reset steps follow those of the standard MEKF implementation.

1.4 Simulated Results

1.4.1 Method to Evaluate Accuracy of Pose Solutions

Two metrics are employed to characterize the performance of the end-to-end relative pose estimate solutions: a "Position Score" and an "Attitude Score". The two scores are defined as follows.

The position score for image i is the 2-norm of the relative position error (difference between the true relative position vector $\boldsymbol{r}^c_{rel,i}$ and the estimated relative position vector $\hat{\boldsymbol{r}}^c_{rel,i}$):

$$pos\_score_i = \|\boldsymbol{r}^c_{rel,i} - \hat{\boldsymbol{r}}^c_{rel,i}\|_2 \quad [m] \tag{1.36}$$

The attitude score for image i is the magnitude of the smallest angle-of-single-rotation that aligns the true and estimated attitudes, expressed in quaternion form ($\boldsymbol{q}_i$ and $\hat{\boldsymbol{q}}_i$, respectively):

$$\delta\boldsymbol{q}_i = \boldsymbol{q}_i \otimes \hat{\boldsymbol{q}}_i^{-1} \tag{1.37}$$

$$\delta q_i^w = \text{ scalar component of } \delta\boldsymbol{q}_i \tag{1.38}$$

$$att\_score_i = 2\arccos\left(|\delta q_i^w|\right) \quad [rad] \tag{1.39}$$

1.4.2 Target Body

For these simulated results, the Cygnus Enhanced cargo resupply ship is selected as the target body (see Fig. 1.5), as the Cygnus vehicle was the target for the Seeker-1 mission. Five specific feature points on Cygnus are selected to be tracked:

- the centroids of each solar panel, indicated by green dots (2)
- the centroid of the payload bus, indicated by the cyan curly brace (1)
- the centroid of each logo on the payload bus, indicated by violet dots (2)

These five points are selected because they are easily identifiable by a CNN trained to identify objects and their locations in an image frame. Note

that the first three feature points are co-planar, which introduces an attitude ambiguity. To help address this problem, the two logos on the payload bus of Cygnus are also selected as feature points because they are out-of-plane with respect to the first three points. In some tests presented in this work, a sixth feature point is also employed: the overall centroid of the entire Cygnus vehicle.

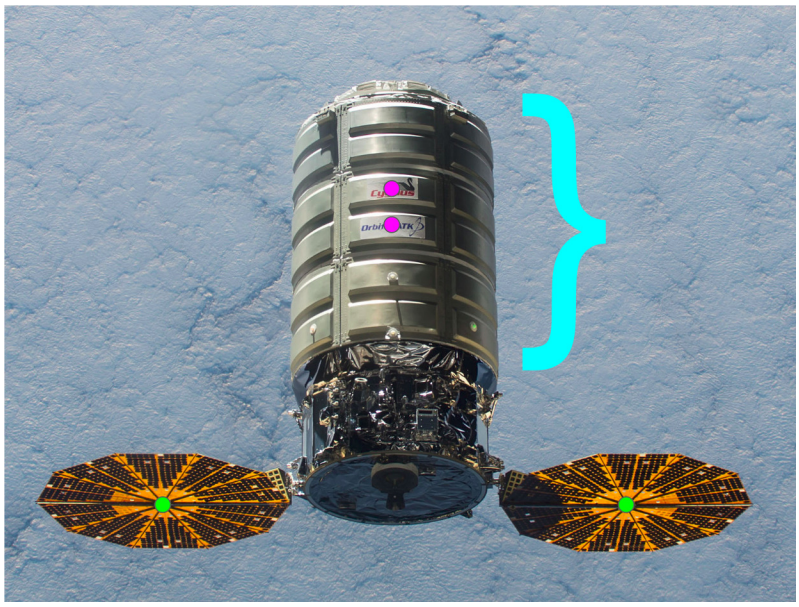


Figure 1.5: Cygnus Enhanced Vehicle (Credit: NASA)

1.4.3 Run-Time Performance Test

The run-time performance of both the NLS and PnP solvers is tested on two different processors: (1) a laptop equipped with a dual-core Intel Core i7 (6th Generation) 6500U rated at 2.5 GHz per core, and (2) an Intel Joule equipped with a quad-core Intel Core i5 (7th Generation) 7260U rated at 2.2

GHz per core. The Intel Joule employed for this test is similar to the processor used on-board Seeker-1.

This test involved 500 randomly generated poses, with five feature points (see Sec. 1.4.2), and five attitude initializations for the NLS solver. The results are summarized below in Tables 1.1 and 1.2 for the NLS and PnP results, respectively. From this test, it is apparent that the NLS solver runs approximately five times faster on the laptop and ten times faster on the Joule, than the PnP solver. However, the average 1 ms run-time per pose solution by the slower PnP solver on the slower Joule ($\approx$ 1 kHz) is still much faster than measurements would reasonably be expected to arrive at. Therefore, the time footprint of the pose solution is negligible compared to that of upstream processes.

Table 1.1: NLS Run-Time Results

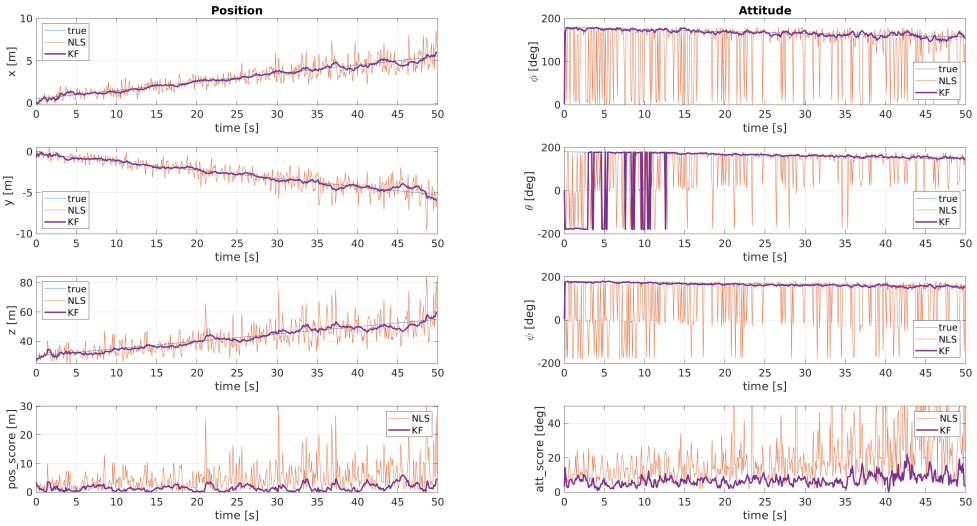
	Average Run-Time	**Total Run-Time**
Laptop	$\approx$ 0.07 ms	$\approx$ 35 ms
Joule	$\approx$ 0.1 ms	$\approx$ 50 ms

Table 1.2: PnP Run-Time Results

	Average Run-Time	**Total Run-Time**
Laptop	$\approx$ 0.36 ms	$\approx$ 180 ms
Joule	$\approx$ 1 ms	$\approx$ 500 ms

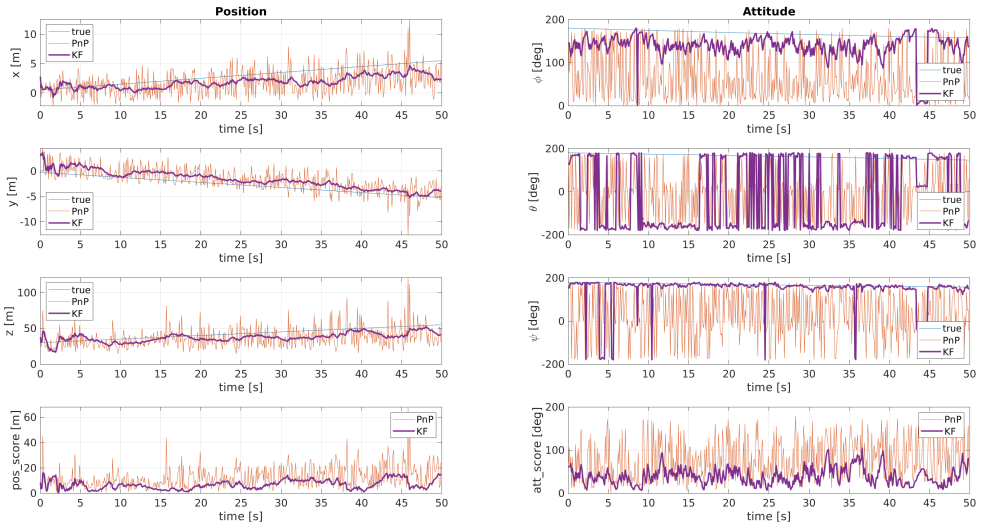
1.4.4 Pose Tracking Test 1

The estimation performance is tested with a simple simulated trajectory that consists of 500 poses. The filter dynamics and measurement receipt rate are both set to 10 Hz. Relative bearing measurements (α, ε) are simulated for six feature points: the five mentioned earlier in addition to the overall Cygnus centroid. The simulated relative bearing measurements are artificially corrupted by additive i.i.d. noise distributed as $\mathcal{N}(0°, 1°)$. Results are provided below for both NLS (Fig. 1.6) and PnP (Fig. 1.7) as the underlying pose solution to the filters.



(a) Position (b) Attitude

Figure 1.6: Pose Tracking Test 1 - NLS



(a) Position (b) Attitude

Figure 1.7: Pose Tracking Test 1 - PnP

The corresponding statistics over time are provided in Tables 1.3 and 1.4, respectively.

Table 1.3: Pose Tracking Test 1 : NLS-Backend Statistics

	Mean	**Std.**
Pos. Score	1.3 m	1.0 m
Att. Score	7.0°	3.0°

Table 1.4: Pose Tracking Test 1 : PnP-Backend Statistics

	Mean	**Std.**
Pos. Score	5.0 m	4.0 m
Att. Score	36.0°	19.0°

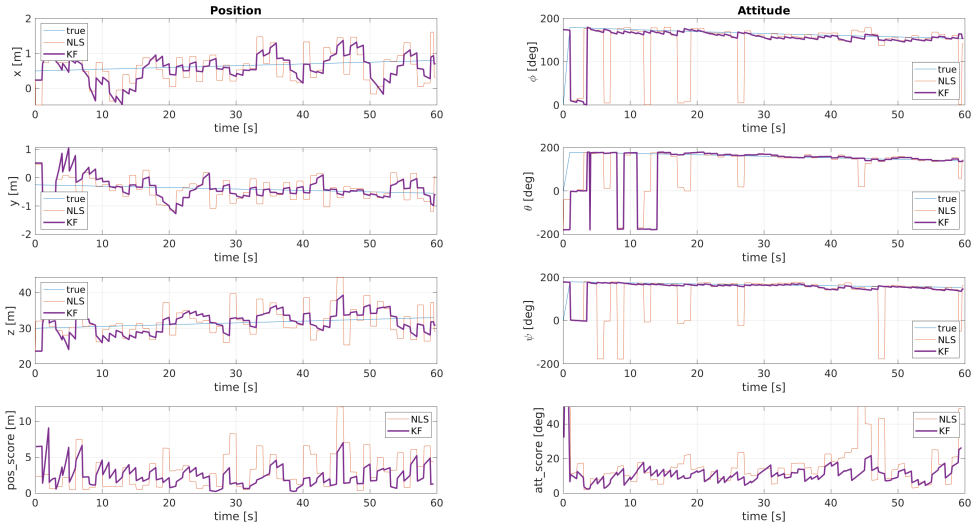
From these results, it is apparent that in addition to being faster, the

NLS solver is also more accurate than the PnP solver, indicated by the lower mean position and attitude scores, and smaller standard deviations over time.

It is important to note that the steep jumps in the filtered estimates for the ϕ, θ, ψ Euler angles in Fig. 1.6b and 1.7b are due to the angle wrap-around problem with the Euler angle representation of attitude, and not due to the actual filtered estimates varying rapidly. Hence, the corresponding filtered attitude scores in the same plots do not exhibit the same rapid changes as seen in the Euler angle estimates.

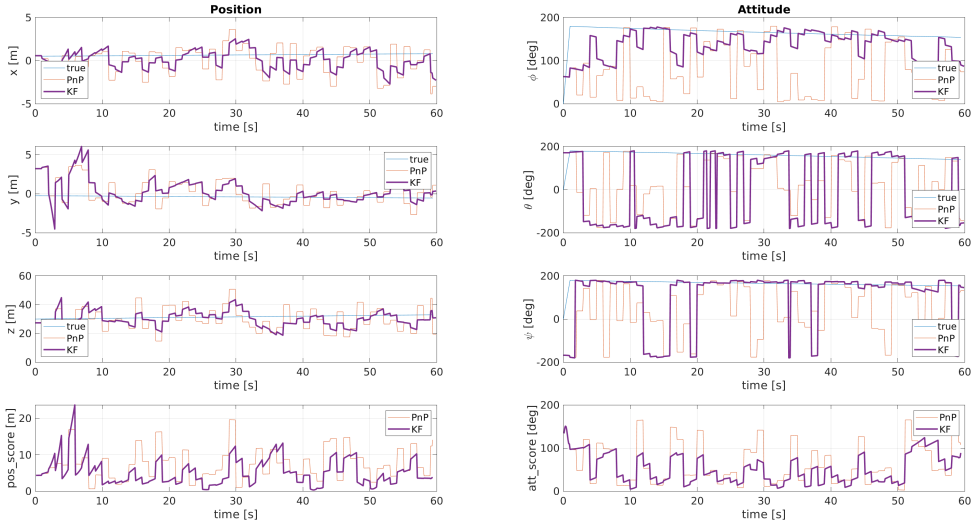
1.4.5 Pose Tracking Test 2

The estimation performance is also tested with a simple simulated trajectory for 60 seconds in a more flight-like set-up. The filter dynamics are set to 10 Hz and the measurement receipt rate is set to 1 Hz. Relative bearing measurements (α, ε) are simulated for the same six feature points as in the previous test. However, the relative bearing measurements are simulated by a different computer process, and communicated to the main process via a named pipe whenever available. The simulated measurements are artificially corrupted by additive i.i.d. noise distributed as $\mathcal{N}(0^\circ,\ 1^\circ)$. Results are provided below for both NLS (Fig. 1.8) and PnP (Fig. 1.9) as the underlying pose solution to the filters.



(a) Position (b) Attitude

Figure 1.8: Pose Tracking Test 2 - NLS



(a) Position (b) Attitude

Figure 1.9: Pose Tracking Test 2 - PnP

Table 1.5: Pose Tracking Test 2 : NLS-Backend Statistics

	Mean	**Std.**
Pos. Score	2.5 m	2.0 m
Att. Score	11.0°	6.0°

Table 1.6: Pose Tracking Test 2 : PnP-Backend Statistics

	Mean	**Std.**
Pos. Score	6.0 m	4.0 m
Att. Score	45.0°	26.0°

Again, the NLS solver is more accurate than the PnP solver; in fact, the difference in performance is even more apparent when measurements are available at a slower rate than the rate at which the internal dynamics are running.

For both pose tracking tests, while the NLS solver is more accurate than the PnP one, there is still room for improvement. The sources of error are primarily due to: (1) the small number of selected feature points and their symmetric relative geometry, and (2) the relatively high noise superimposed on the simulated relative bearing measurements. If more feature points are tracked by the upstream CNN, and the CNN is able to provide the relative bearing estimates with higher accuracy, then the accuracy of the downstream pose solution will naturally be better.

1.5 Conclusion

In this chapter, an open-source, general-purpose pose estimation software package written in C++ has been presented, which ingests relative bearing (az., el.) measurements to *a priori* determined 3D feature points on a target body from a monocular camera, and provides a filtered full relative pose solution for the target with respect to the chaser. The software package implements both a non-linear least-squares and a perspective-n-point formulation for the underlying pose solution. The pose filtering is accomplished with a standard Linear Kalman Filter for position and a Multiplicative Extended Kalman Filter for attitude. The software has been designed to run in flight-like conditions, with support for measurement input and filtered solution output via serialized data streams over named pipes, and some basic memory management and logging.

In addition, simulated results for NASA Johnson Space Center led Seeker program have been provided to demonstrate the capabilities of the software package for a real mission scenario. A basic performance test has been conducted to compare the run-time performances on a representative laptop computer, and a representative cubesat flight computer (the Intel Joule). Finally, two simulated pose tracking tests are conducted to demonstrate the software package.

Several avenues for future work exist. First, to better exploit all the information available, the capability to propagate the relative bearing measurement noise through the NLS solver to feed into the filter is currently being

developed. Second, to increase the accuracy of the underlying pose estimation, incorporating more feature points that are also better spatially distributed across the Cygnus spacecraft is being investigated. Finally, a more robust system identification to more accurately characterize the noise and kinematic model would help improve performance.

Chapter 2

Angular Velocity Estimation of a Rigid-Body in Near Pure-Spin Condition

2.1 Introduction

This chapter of the report describes a batch estimator based on a time-series of relative orientation measurements at uniform intervals to estimate a target object's spin rate, spin-axis direction, and under certain simplifying assumptions, the associated spin-axis direction estimate covariance matrix. In addition, the estimator also provides variance estimates for the spin rate, and for the two Euler angles parameterizing the spin-axis direction. An earlier version of this work was presented at the 2020 AAS/AIAA Astrodynamics Specialist Conference (AAS 20-652) [14]. When dealing with non-cooperative targets, such as asteroids or out-of-commission spacecraft, the target object's inertia properties and/or any external torques applied upon the same are often unknown. Therefore, a kinematic approach has certain clear advantages for these settings as we pursue in this work. Instead of the commonly-used Multiplicative Extended Kalman Filter (MEKF) formulation often taken to solve attitude and angular velocity estimation problems [10, 12, 15], we exploit the problem's underlying geometry to develop a batch estimator that brings forth certain powerful mathematical results associated with linear algebra and the

singular value decomposition (SVD). Importantly, this work introduces explicit analytical expressions for the singular values and singular vectors of the measurement matrix. In this context, we note that this work can be viewed to be a generalization for some preliminary results of Mortari and Akella [16] and the recent QuateRA algorithm [13]. The QuateRA algorithm, specifically, provides an optimal estimate for the angular velocity magnitude and spin-axis direction. Though QuateRA does provide an accompanying covariance for the overall angular velocity vector (the spin-axis direction scaled by the angular velocity magnitude), the algorithm resorts to an information-based (MEKF-like), recursive covariance estimation process. This new work fills the gap by providing an explicit, non-recursive covariance bound for the estimated spin-axis direction by itself without the need for any additional iterations.

The formulation presented in this work restricts itself to the case of rigid-bodies undergoing constant-rate pure-spin, which is a special case of the general problem of relative angular velocity estimation for arbitrarily tumbling bodies. However, any tumbling motion, under sufficiently short duration time intervals, can be reasonably approximated as pure-spin about an instantaneous spin-axis direction. Furthermore, the geometric properties uncovered through this work provide clear indications for when the pure-spin assumption is sufficiently violated, and accordingly, to inform an adaptive sliding-window batch size selection for the pure-spin assumption to be valid again [16].

The remainder of the chapter is organized as follows. First, the problem is posed and the theoretical developments are presented in the Problem

Formulation section. Next, the developed algorithm to estimate the spin-axis direction and the associated covariance matrix is summarized in the Algorithm Outline section. Results from several Monte Carlo simulations are provided in the Results section. Finally, concluding remarks and notes for future work are made in the Conclusion section.

2.2 Problem Formulation

2.2.1 Noise-Free Measurements

Orientation measurements of a target body in pure-spin about some arbitrary, but fixed, spin-axis are assumed to be obtained over uniformly-spaced time-steps ($k = 1, 2, \ldots$). The true target rotation, assumed to be undergoing pure-spin about a fixed axis by a constant rate, is modeled as a 3-1-3 Euler angle rotation sequence parameterized by the in-plane angle $k\theta$, the constant obliquity angle β, and the constant precession angle γ. The angle θ by itself represents the (constant) in-plane rotation per time-step.

Therefore, the noise-free measurement model at time $t = t_k$ is expressed as:

$$\bar{\boldsymbol{z}}_k = R_3(\gamma)R_1(\beta)R_3(k\theta)\hat{\boldsymbol{e}}_1 \tag{2.1}$$

$$= \begin{bmatrix} \cos(k\theta)\cos(\gamma) - \sin(k\theta)\cos(\beta)\sin(\gamma) \\ \cos(k\theta)\sin(\gamma) + \sin(k\theta)\cos(\beta)\cos(\gamma) \\ \sin(k\theta)\sin(\beta) \end{bmatrix} \tag{2.2}$$

$$= \cos(k\theta)\underbrace{\begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \\ 0 \end{bmatrix}}_{\hat{\boldsymbol{\nu}}_1} + \sin(k\theta)\underbrace{\begin{bmatrix} -\cos(\beta)\sin(\gamma) \\ \cos(\beta)\cos(\gamma) \\ \sin(\beta) \end{bmatrix}}_{\hat{\boldsymbol{\nu}}_2} \tag{2.3}$$

where $\hat{\boldsymbol{e}}_1 := \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. The unit vectors $\hat{\boldsymbol{\nu}}_1$ and $\hat{\boldsymbol{\nu}}_2$ are orthogonal and span the plane-of-rotation. Thus, the cross-product:

$$\hat{\boldsymbol{n}} = \hat{\boldsymbol{\nu}}_1 \times \hat{\boldsymbol{\nu}}_2 \tag{2.4}$$

$$= \begin{bmatrix} \sin(\beta)\sin(\gamma) \\ -\sin(\beta)\cos(\gamma) \\ \cos(\beta) \end{bmatrix} \tag{2.5}$$

defines the normal direction $\hat{\boldsymbol{n}}$ to the plane-of-rotation; that is, the spin-axis direction. The geometry of the problem is depicted in Fig. 2.1, with measurement $\bar{\boldsymbol{z}}_k$, cumulative in-plane rotation $k\theta$, obliquity angle β, and spin-direction axis $\hat{\boldsymbol{n}}$ labeled. The precession angle γ is set to zero in this depiction (for purposes of pictorial simplicity), and thus not labeled.

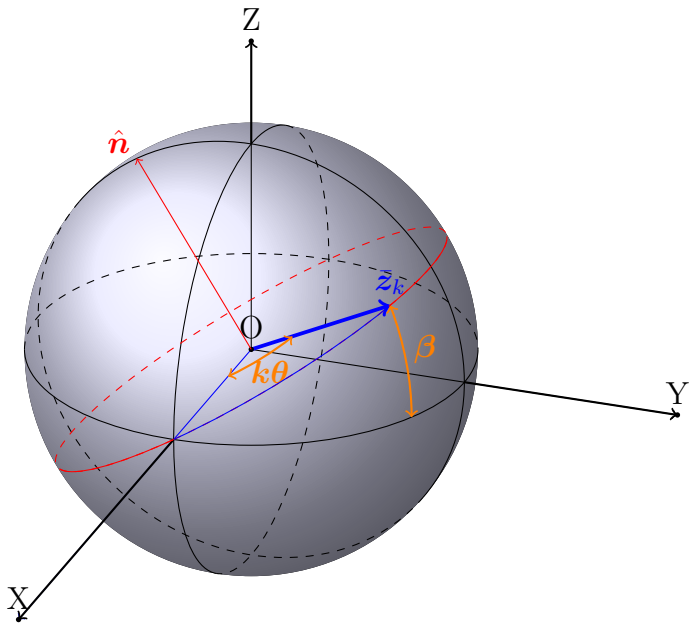


Figure 2.1: Problem Geometry

Measurements $\bar{\boldsymbol{z}}_k$ are stacked up from time t_0 to t_n into a measurement matrix $\bar{Z}_{n+1}$, where $(n+1)$ is the total number of measurements:

$$\bar{Z}_{n+1} = \begin{bmatrix} \bar{\boldsymbol{z}}_0 & \bar{\boldsymbol{z}}_1 & \dots & \bar{\boldsymbol{z}}_n \end{bmatrix}_{3\times(n+1)} \tag{2.6}$$

Consider the noise-free spatiotemporal matrix $\bar{Q}_{n+1} := \bar{Z}_{n+1}\bar{Z}_{n+1}^T$, which is a real, symmetric, positive semi-definite 3×3 matrix. The eigenvalues and the eigenvectors of $\bar{Q}_{n+1}$ therefore coincide with the squared singular values and the left singular vectors of $\bar{Z}_{n+1}$, respectively. The eigen and singular value decompositions can be respectively expressed as:

$$\bar{Q}_{n+1} = \bar{U}\bar{\Lambda}\bar{U}^T \tag{2.7}$$

$$\bar{Z}_{n+1} = \bar{U}\bar{\Sigma}\bar{V}^T = \sum_{j=1}^{3}(\bar{\boldsymbol{u}}_j)\bar{s}_j(\bar{\boldsymbol{v}}_j)^T \tag{2.8}$$

where in particular:

$$\bar{U} = \begin{bmatrix}\bar{\boldsymbol{u}}_1 & \bar{\boldsymbol{u}}_2 & \bar{\boldsymbol{u}}_3\end{bmatrix} \tag{2.9}$$

$$\bar{V} = \begin{bmatrix}\bar{\boldsymbol{v}}_1 & \bar{\boldsymbol{v}}_2 & \bar{\boldsymbol{v}}_3\end{bmatrix} \tag{2.10}$$

$$\bar{\Lambda} = \begin{bmatrix}\bar{\lambda}_1 & 0 & 0 \\ 0 & \bar{\lambda}_2 & 0 \\ 0 & 0 & \bar{\lambda}_3\end{bmatrix} \tag{2.11}$$

$$\bar{\Sigma} = \begin{bmatrix}\bar{s}_1 & 0 & 0 \\ 0 & \bar{s}_2 & 0 \\ 0 & 0 & \bar{s}_3\end{bmatrix} = \begin{bmatrix}\sqrt{\bar{\lambda}_1} & 0 & 0 \\ 0 & \sqrt{\bar{\lambda}_2} & 0 \\ 0 & 0 & \sqrt{\bar{\lambda}_3}\end{bmatrix} \tag{2.12}$$

The left singular vectors of $\bar{Z}_{n+1}$ coincide with the eigenvectors of $\bar{Q}_{n+1}$, and are denoted by $\bar{\boldsymbol{u}}_i$ in Eq. 2.9 for $i \in \{1, 2, 3\}$. The right singular vectors of $\bar{Z}_{n+1}$ are represented by $\bar{\boldsymbol{v}}_i$ in Eq. 2.10 for $i \in \{1, 2, 3\}$, but they are not particularly relevant to the results obtained in this work. The eigenvalues of $\bar{Q}_{n+1}$ are notated by $\bar{\lambda}_i$ in Eq. 2.11 for $i \in \{1, 2, 3\}$. The singular values of $\bar{Z}_{n+1}$ are notated by $\bar{s}_i = \sqrt{\bar{\lambda}_i}$ in Eq. 2.12 for $i \in \{1, 2, 3\}$.

One of the important contributions of this work is the explicit expressions for the three eigenvalues of the matrix $\bar{Q}_{n+1}$. The eigenvalue expressions

depend on whether $(n+1)$ is an even or odd positive integer. For the case when $(n+1)$ is even, the three eigenvalues are given by:

$$\begin{aligned}
\bar{\lambda}_1 &= \frac{n+1}{2} + \sum_{k=0}^{(n-1)/2} \cos\Big((2k+1)\theta\Big) \\
\bar{\lambda}_2 &= \frac{n+1}{2} - \sum_{k=0}^{(n-1)/2} \cos\Big((2k+1)\theta\Big) \\
\bar{\lambda}_3 &= 0
\end{aligned} \tag{2.13}$$

On the other hand, when $(n+1)$ is odd, the following expressions hold for the three eigenvalues of the matrix $\bar{Q}_{n+1}$:

$$\begin{aligned}
\bar{\lambda}_1 &= \frac{n}{2} + \sum_{k=0}^{n/2} \cos\Big(2k\theta\Big) \\
\bar{\lambda}_2 &= \frac{n}{2} + 1 - \sum_{k=0}^{n/2} \cos\Big(2k\theta\Big) \\
\bar{\lambda}_3 &= 0
\end{aligned} \tag{2.14}$$

Note that the eigenvalues expressions in Eqs. 2.13, 2.14 are periodic due to the sums of cosines. For zero measurements $(n+1=0)$, all three eigenvalues coincide: $\big(\bar{\lambda}_i = 0\ ;\ i \in \{1,2,3\}\big)$. For $n\theta \in (0,\pi)$, we have that $\bar{\lambda}_1 > \bar{\lambda}_2$. That is, the expressions in Eqs. 2.13, 2.14 maintain descending order for $n\theta \in (0,\pi)$. When $n\theta = \pi$, then the first two eigenvalues ($\bar{\lambda}_1$ and $\bar{\lambda}_2$) coincide again; that is, they are repeated eigenvalues. For $n\theta \in (\pi, 2\pi)$, the first two eigenvalues swap places: $\bar{\lambda}_2 > \bar{\lambda}_1$. Therefore, care must be taken to ensure which eigenvalue is labeled as $\bar{\lambda}_1$ and which eigenvalue is labeled

as $\bar{\lambda}_2$ when numerically performing the eigen decomposition or the singular value decomposition. It is important to note that this eigenvalue swapping occurs under two situations: (1) when the number of measurements in the batch $(n+1)$ grows large, or (2) when measurements are sparse (θ is large), to qualitatively indicate fast rotations.

These eigenvalue expressions (Eqs. 2.13, 2.14) are derived following the fact that the matrices $\bar{Q}_{n+1} = \bar{Z}_{n+1}\bar{Z}_{n+1}^T$ and $\bar{Z}_{n+1}^T\bar{Z}_{n+1}$ share the same non-zero eigenvalues. Pertinent here is also the fact that $\bar{Z}_{n+1}^T\bar{Z}_{n+1}$ is a symmetric Toeplitz matrix with a recursive form:

$$\begin{aligned}\bar{Z}_{n+1}^T\bar{Z}_{n+1} &= \begin{bmatrix} 1 & \cos(\theta) & \cos(2\theta) & \dots & \cos(n\theta) \\ \cos(\theta) & 1 & \cos(\theta) & \dots & \cos\left((n-1)\theta\right) \\ \cos(2\theta) & \cos(\theta) & 1 & \dots & \cos\left((n-2)\theta\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \cos\left(n\theta\right) & \cos\left((n-1)\theta\right) & \cos\left((n-2)\theta\right) & \dots & 1 \end{bmatrix}_{(n+1)\times(n+1)} \\ &= \begin{bmatrix} \begin{bmatrix} \bar{Z}_n^T\bar{Z}_n \end{bmatrix}_{n\times n} & \boldsymbol{t} \\ \boldsymbol{t}^T & 1 \end{bmatrix}_{(n+1)\times(n+1)} \end{aligned} \tag{2.15}$$

where $\boldsymbol{t}^T = \begin{bmatrix}\cos\left(n\theta\right) & \cos\left((n-1)\theta\right) & \cos\left((n-2)\theta\right) & \dots & \cos\left(\theta\right)\end{bmatrix}_{1\times n}$.

The explicit eigenvalue expressions listed in Eqs. 2.13, 2.14 are thus obtained following the application of Thm. 3.1 of Melman [17] upon the recursive symmetric Toeplitz form of $\bar{Z}_{n+1}^T\bar{Z}_{n+1}$ (Eq. 2.15).

In particular, Melman's Thm. 3.1 applied to this problem yields the eigenvalues $\left(\bar{\lambda}\right)$ of $\bar{Z}_{n+1}^T\bar{Z}_{n+1}$ that are not also eigenvalues of $\bar{Z}_n^T\bar{Z}_n$, whose

associated eigenspaces are not entirely contained in $\{\boldsymbol{t}\}^{\perp}$. These eigenvalues $\bar{\lambda}$ are given by the solutions to the equation:

$$-1+\bar{\lambda}+\boldsymbol{t}^{T}\left(\bar{Z}_{n}^{T}\bar{Z}_{n}-\bar{\lambda}I_{n\times n}\right)^{-1}\boldsymbol{t}=0 \tag{2.16}$$

This relationship can be derived from the matrix partitioning presented in Eq. 2.15. The satisfaction of the eigenspace condition can be inferred from the recursive construction of the Toeplitz structure in Eq. 2.15 based on new measurements being stacked into $\bar{Z}_n$. Therefore, Eq. 2.16 is recursively applied to generate the eigenvalues of $\bar{Z}_{n+1}^{T}\bar{Z}_{n+1}$, as given in Eqs. 2.13, 2.14.

Next, proceeding with computation of the left singular vectors, the first left singular vector of $\bar{Z}_{n+1}$ is given by:

$$\bar{\boldsymbol{u}}_1=\begin{bmatrix}\cos\left(\frac{n}{2}\theta\right)\cos(\gamma)-\sin\left(\frac{n}{2}\theta\right)\cos(\beta)\sin(\gamma)\\ \cos\left(\frac{n}{2}\theta\right)\sin(\gamma)+\sin\left(\frac{n}{2}\theta\right)\cos(\beta)\cos(\gamma)\\ \sin\left(\frac{n}{2}\theta\right)\sin(\beta)\end{bmatrix} \tag{2.17}$$

$$=\cos\left(\frac{n}{2}\theta\right)\underbrace{\begin{bmatrix}\cos(\gamma)\\ \sin(\gamma)\\ 0\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_1}+\sin\left(\frac{n}{2}\theta\right)\underbrace{\begin{bmatrix}-\cos(\beta)\sin(\gamma)\\ \cos(\beta)\cos(\gamma)\\ \sin(\beta)\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_2} \tag{2.18}$$

The second left singular vector is given by:

$$\bar{\boldsymbol{u}}_2=\begin{bmatrix}-\sin\left(\frac{n}{2}\theta\right)\cos(\gamma)-\cos\left(\frac{n}{2}\theta\right)\cos(\beta)\sin(\gamma)\\ -\sin\left(\frac{n}{2}\theta\right)\sin(\gamma)+\cos\left(\frac{n}{2}\theta\right)\cos(\beta)\cos(\gamma)\\ \cos\left(\frac{n}{2}\theta\right)\sin(\beta)\end{bmatrix} \tag{2.19}$$

$$=-\sin\left(\frac{n}{2}\theta\right)\underbrace{\begin{bmatrix}\cos(\gamma)\\ \sin(\gamma)\\ 0\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_1}+\cos\left(\frac{n}{2}\theta\right)\underbrace{\begin{bmatrix}-\cos(\beta)\sin(\gamma)\\ \cos(\beta)\cos(\gamma)\\ \sin(\beta)\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_2} \tag{2.20}$$

The third left singular vector $\bar{\boldsymbol{u}}_3$ is the cross-product between the first two left singular vectors: $\bar{\boldsymbol{u}}_3 = \bar{\boldsymbol{u}}_1 \times \bar{\boldsymbol{u}}_2$.

The above singular vector expressions are symbolically derived from $\bar{Q}_{n+1} = \bar{Z}_{n+1}\bar{Z}_{n+1}^T$ for varying number of measurements $(n+1)$, and an induction argument is made to arrive at the expressions in Eqs. 2.17, 2.19. By substituting the expressions for left singular vectors $\bar{\boldsymbol{u}}_i$ and squared singular values $\bar{s}_i^2 = \bar{\lambda}_i$, where $i \in \{1, 2, 3\}$, into Eq. 2.9 and Eq. 2.11, respectively, we can verify that the above left singular vector and singular value expressions indeed hold via Eq. 2.7.

Comparing the expression for $\bar{\boldsymbol{u}}_1$ (Eq. 2.18) with the expression for $\bar{\boldsymbol{z}}_k$ (Eq. 2.3), it is clear that the first left singular vector $\bar{\boldsymbol{u}}_1$ of the measurement matrix $\bar{Z}_{n+1}$ qualitatively represents the *mean* rotation $\left(\frac{n}{2}\theta\right)$ in the plane-of-rotation over the time-steps compiled.

2.2.2 Noisy Measurements

Now, consider the case when measurements $\bar{\boldsymbol{z}}_k$ are corrupted by time-uncorrelated measurement noise (in multiplicative form) at each time $t = t_k$,

represented by noise rotation matrix $R_\eta(k)$:

$$\boldsymbol{z}_k = R_\eta(k)\bar{\boldsymbol{z}}_k \tag{2.21}$$

$$= R_\eta(k)R_3(\gamma)R_1(\beta)R_3(k\theta)\hat{e}_1 \tag{2.22}$$

$$= R_\eta(k)\begin{bmatrix}\cos(k\theta)\cos(\gamma) - \sin(k\theta)\cos(\beta)\sin(\gamma)\\ \cos(k\theta)\sin(\gamma) + \sin(k\theta)\cos(\beta)\cos(\gamma)\\ \sin(k\theta)\sin(\beta)\end{bmatrix} \tag{2.23}$$

$$= \cos(k\theta)R_\eta(k)\underbrace{\begin{bmatrix}\cos(\gamma)\\ \sin(\gamma)\\ 0\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_1} + \sin(k\theta)R_\eta(k)\underbrace{\begin{bmatrix}-\cos(\beta)\sin(\gamma)\\ \cos(\beta)\cos(\gamma)\\ \sin(\beta)\end{bmatrix}}_{\hat{\boldsymbol{\nu}}_2} \tag{2.24}$$

Just as in the earlier noise-free measurements case, measurements (Eq. 2.21) are once again stacked up from time t_0 to t_n into a measurement matrix Z_{n+1}, where $(n+1)$ is once again the total number of measurements:

$$Z_{n+1} = \begin{bmatrix}R_\eta(0)\bar{\boldsymbol{z}}_0 & R_\eta(1)\bar{\boldsymbol{z}}_1 & \dots & R_\eta(n)\bar{\boldsymbol{z}}_n\end{bmatrix}_{3\times(n+1)} \tag{2.25}$$

$$= \begin{bmatrix}\boldsymbol{z}_0 & \boldsymbol{z}_1 & \dots & \boldsymbol{z}_n\end{bmatrix}_{3\times(n+1)} \tag{2.26}$$

Consider the noisy spatiotemporal matrix $Q_{n+1} := Z_{n+1}Z_{n+1}^T$, which is a real, symmetric, positive semi-definite 3×3 matrix. The eigenvalues and the eigenvectors of Q_{n+1} therefore coincide with the squared singular values and the left singular vectors of Z_{n+1}, respectively. As demonstrated by Mortari and Akella [16], the first two left singular vectors of Z_{n+1} (labeled respectively as $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$) span the least-squares-fit plane-of-rotation. Therefore, the *best* least-squares estimate of the spin-axis $\hat{\boldsymbol{n}}$ is given by the cross-product of the noisy first two left singular vectors:

$$\hat{\boldsymbol{n}} = \boldsymbol{u}_1 \times \boldsymbol{u}_2 \tag{2.27}$$

Assuming the body to be undergoing pure-spin rotation by a constant rate, the best current estimates at time $t = t_n$ for the Euler angles θ, β, and γ can be extracted from the first two left singular vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ of the noisy measurement matrix Z_{n+1}, and their cross-product $\hat{\boldsymbol{n}} = \boldsymbol{u}_1 \times \boldsymbol{u}_2$, where we assume the forms of the first two left singular vectors to be those of the noise-free cases $\bar{\boldsymbol{u}}_1$ (in Eq. 2.17) and $\bar{\boldsymbol{u}}_2$ (in Eq. 2.19), respectively, and $\hat{\boldsymbol{n}}$ of the noise-free form in Eq. 2.4.

In particular, the angle estimates $\hat{\theta}$, $\hat{\beta}$, $\hat{\gamma}$ can be extracted as:

$$\begin{aligned}\hat{\theta} &= \frac{2}{n}\sin^{-1}\left(\frac{\boldsymbol{u}_1(3)}{\sin(\hat{\beta})}\right) = \frac{2}{n}\sin^{-1}\left(\frac{\boldsymbol{u}_1(3)}{\sqrt{1-\left(\hat{\boldsymbol{n}}(3)\right)^2}}\right) \\ &= \frac{2}{n}\cos^{-1}\left(\frac{\boldsymbol{u}_2(3)}{\sin(\hat{\beta})}\right) = \frac{2}{n}\cos^{-1}\left(\frac{\boldsymbol{u}_2(3)}{\sqrt{1-\left(\hat{\boldsymbol{n}}(3)\right)^2}}\right)\end{aligned} \tag{2.28}$$

$$\hat{\beta} = \cos^{-1}\Big(\hat{\boldsymbol{n}}(3)\Big) \tag{2.29}$$

$$\hat{\gamma} = \tan^{-1}\left(\frac{-\hat{\boldsymbol{n}}(1)}{\hat{\boldsymbol{n}}(2)}\right) \tag{2.30}$$

Note that the above angle expressions are not unique; there exist many trigonometric alternatives for obtaining the angle estimates from $\bar{\boldsymbol{u}}_1$, $\bar{\boldsymbol{u}}_2$, and

$\hat{\boldsymbol{n}}$. Depending on the inverse trigonometric function employed, it is natural to expect quadrant ambiguity issues to arise, which must be carefully accounted for during implementation. Furthermore, the various expressions for the same angle estimate have different sensitivities to the singular vector computations under measurement noise, depending on the nonlinear function employed.

2.2.3 Covariance Estimate for the Spin-Axis Direction

To derive an approximation for the covariance (error bounds) for the spin-axis direction estimate, certain simplifying assumptions are made. First, assume the noise rotation matrix $R_\eta(k)$ at each time $t = t_k$ is parameterized by the axis-angle representation $(\hat{\boldsymbol{e}}_k,\ \phi_k)$. The corresponding Gibbs vector representation is given by:

$$\boldsymbol{g}_k = \hat{\boldsymbol{e}}_k \tan(\phi_k/2) = \boldsymbol{a}_k^g/2 \rightarrow \boldsymbol{a}_k^g = 2\boldsymbol{g}_k \tag{2.31}$$

with magnitude $a_k^g = \|\boldsymbol{a}_k^g\|_2$. The equivalent rotation matrix, which is the attitude error matrix, is given exactly by [18]:

$$R_\eta(k) = R\left(\boldsymbol{a}_k^g\right) = \frac{1}{1+\frac{1}{4}\left(a_k^g\right)^2}\left(\left(1-\frac{1}{4}\left(a_k^g\right)^2\right) I_{3\times 3} + \frac{1}{2}\left(\boldsymbol{a}_k^g\right)\left(\boldsymbol{a}_k^g\right)^T - \left[\boldsymbol{a}_k^g\times\right]\right) \tag{2.32}$$

where the skew-symmetric cross-product matrix for arbitrary vector $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ is defined by:

$$[\boldsymbol{v}\times] := \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{2.33}$$

To second-order, the equivalent rotation matrix is given by [10]:

$$R_\eta(k) = R\left(\boldsymbol{a}_k^g\right) \approx I_{3\times 3} - [\boldsymbol{a}_k^g\times] - \frac{1}{2}\left(\left(a_k^g\right)^2 I_{3\times 3} - \left(\boldsymbol{a}_k^g\right)\left(\boldsymbol{a}_k^g\right)^T\right) \tag{2.34}$$

Finally, the linear approximation of the equivalent rotation matrix is given by:

$$R_\eta(k) = R\left(\boldsymbol{a}_k^g\right) \approx I_{3\times 3} - [\boldsymbol{a}_k^g\times] \tag{2.35}$$

Therefore, the noisy measurement matrix Z_{n+1} (Eq. 2.25) can be approximated to first-order as:

$$Z_{n+1} = \begin{bmatrix} R_\eta(0)\bar{\boldsymbol{z}}_0 & R_\eta(1)\bar{\boldsymbol{z}}_1 & \dots & R_\eta(n)\bar{\boldsymbol{z}}_n \end{bmatrix}_{3\times(n+1)} \tag{2.36}$$

$$\approx \begin{bmatrix} \left(I - [\boldsymbol{a}_0^g\times]\right)\bar{\boldsymbol{z}}_0 & \left(I - [\boldsymbol{a}_1^g\times]\right)\bar{\boldsymbol{z}}_1 & \dots & \left(I - [\boldsymbol{a}_n^g\times]\right)\bar{\boldsymbol{z}}_n \end{bmatrix}_{3\times(n+1)} \tag{2.37}$$

$$= \underbrace{\begin{bmatrix} \bar{\boldsymbol{z}}_0 & \bar{\boldsymbol{z}}_1 & \dots & \bar{\boldsymbol{z}}_n \end{bmatrix}}_{\bar{Z}_{n+1}} - \underbrace{\begin{bmatrix} [\boldsymbol{a}_0^g\times]\,\bar{\boldsymbol{z}}_0 & [\boldsymbol{a}_1^g\times]\,\bar{\boldsymbol{z}}_1 & \dots & [\boldsymbol{a}_n^g\times]\,\bar{\boldsymbol{z}}_n \end{bmatrix}}_{-\Delta Z_{n+1}} \tag{2.38}$$

$$Z_{n+1} = \bar{Z}_{n+1} + \Delta Z_{n+1} \tag{2.39}$$

which is a sum of the noise-free measurement matrix $\bar{Z}_{n+1}$ (in Eq. 2.6) and a noise-induced additive error matrix ΔZ_{n+1}. Next, we assume

that ΔZ_{n+1} is composed of zero-mean, uncorrelated elements with variance ε^2 $\left(\text{that is, for example: } \Delta Z_{n+1;(i,j)} \sim \mathcal{N}(0, \varepsilon^2)\right)$. Thus, by Epps and Krivitzky [19], the first two moments of each element $i \in \{1, 2, 3\}$ of the three left singular vectors $\boldsymbol{u}_j$ ($j \in \{1, 2, 3\}$) of the noisy measurement matrix Z_{n+1} can be obtained. To help further outline this process, we consider the singular value decomposition:

$$Z_{n+1} = U\Sigma V^T = \sum_{j=1}^{3} (\boldsymbol{u}_j) s_j (\boldsymbol{v}_j)^T \tag{2.40}$$

The mean and standard deviation of the three ($j \in \{1, 2, 3\}$) left singular vectors $\boldsymbol{u}_{j,i}$ by vector element $i \in \{1, 2, 3\}$ are given by:

$$\langle \boldsymbol{u}_{j,i} \rangle = \left(1 - \frac{\varepsilon^2}{2} \sum_{m=1;\ m\neq j}^{3} \frac{\bar{\lambda}_m + \bar{\lambda}_j}{(\bar{\lambda}_m - \bar{\lambda}_j)^2}\right) \bar{\boldsymbol{u}}_{j,i} + \mathcal{O}\left(\varepsilon^4\right) \tag{2.41}$$

$$\sigma_{j,i} = \varepsilon \sqrt{\sum_{m=1;\ m\neq j}^{3} \frac{\bar{\lambda}_m + \bar{\lambda}_j}{(\bar{\lambda}_m - \bar{\lambda}_j)^2} \left(\bar{\boldsymbol{u}}_{m,i}\right)^2} + \mathcal{O}\left(\varepsilon^2\right) \tag{2.42}$$

where $\bar{\lambda}_i$ is the i^{th} eigenvalue of the noise-free spatiotemporal matrix $\bar{Q}_{n+1} = \bar{Z}_{n+1}\bar{Z}_{n+1}^T$ and $\bar{\boldsymbol{u}}_{j,i}$ is the i^{th} element of the j^{th} left singular vector of the noise-free measurement matrix $\bar{Z}_{n+1}$. In this context, we recall the analytical expressions for the aforementioned eigenvalues and left singular vectors for the noise-free case, which have already been derived and presented in Eqs. 2.13, 2.14, 2.17, 2.19.

Next, assume the noise standard deviation $\varepsilon \ll 1$ such that $\langle \boldsymbol{u}_{j,i} \rangle \approx \bar{\boldsymbol{u}}_{j,i}$. Now, construct diagonal covariance matrices for $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ based on the standard deviations $\sigma_{j,i}$ by Eq. 2.42, respectively:

$$P_1 = diag\left(\left(\sigma_{1,1}\right)^2,\ \left(\sigma_{1,2}\right)^2,\ \left(\sigma_{1,3}\right)^2\right) \tag{2.43}$$

$$P_2 = diag\left(\left(\sigma_{2,1}\right)^2,\ \left(\sigma_{2,2}\right)^2,\ \left(\sigma_{2,3}\right)^2\right) \tag{2.44}$$

Define the stacked vector $\boldsymbol{u} := \begin{bmatrix}\boldsymbol{u}_1^T & \boldsymbol{u}_2^T\end{bmatrix}^T$ with associated covariance matrix:

$$P = \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{bmatrix}_{6\times 6} \tag{2.45}$$

Note that expressing the covariance matrix P as in Eq. 2.45 makes the assumption that $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are independent, which is not really true given the necessary orthogonality of the two left singular vectors. However, results from Monte Carlo simulations presented in the sequel qualitatively show that this simplifying assumption does not significantly impact the probability distribution of $\hat{\boldsymbol{n}} = \boldsymbol{u}_1 \times \boldsymbol{u}_2$.

Since the spin-axis direction estimate is given by the cross-product $\hat{\boldsymbol{n}} = \boldsymbol{u}_1 \times \boldsymbol{u}_2$, the cross-product operator can be expressed as a $\mathbb{R}^6 \mapsto \mathbb{R}^3$ transformation from $\boldsymbol{u}$ to $\hat{\boldsymbol{n}}$ with Jacobian:

$$J = \begin{bmatrix} -[\boldsymbol{u}_2\times] & [\boldsymbol{u}_1\times] & \end{bmatrix}_{3\times 6} \tag{2.46}$$

Thus, the resulting covariance matrix of the spin-axis direction estimate $\hat{\boldsymbol{n}}$ is given by:

$$P_{\hat{\boldsymbol{n}}} = JPJ^T \tag{2.47}$$

$$= \begin{bmatrix} -[\boldsymbol{u}_2\times] & [\boldsymbol{u}_1\times] & \end{bmatrix}_{3\times 6} \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{bmatrix}_{6\times 6} \begin{bmatrix} -[\boldsymbol{u}_2\times]^T \\ [\boldsymbol{u}_1\times]^T \end{bmatrix}_{6\times 3} \tag{2.48}$$

$$P_{\hat{\boldsymbol{n}}} = [\boldsymbol{u}_2\times] P_1 [\boldsymbol{u}_2\times]^T + [\boldsymbol{u}_1\times] P_2 [\boldsymbol{u}_1\times]^T \tag{2.49}$$

An alternative to pursuing the cross-product transformation approach presented above is the following. Because the third left singular vector $\boldsymbol{u}_3$ coincides with the spin-axis direction estimate $\hat{\boldsymbol{n}}$, the covariance matrix $P_{\hat{\boldsymbol{n}}}$ can be directly computed by constructing the diagonal covariance matrix for $\boldsymbol{u}_3$ based on the standard deviations $\sigma_{3,i}$ by Eq. 2.42:

$$P_{\hat{\boldsymbol{n}}} = P_3 = diag\left(\left(\sigma_{3,1}\right)^2,\ \left(\sigma_{3,2}\right)^2,\ \left(\sigma_{3,3}\right)^2\right) \tag{2.50}$$

Importantly, this approach does not assume that $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are independent. Either the expression in Eq. 2.47 or the one in Eq. 2.50 may be employed to compute $P_{\hat{\boldsymbol{n}}}$.

Another important point to note is the following: the assumption in Eq. 2.39 that ΔZ_{n+1} is composed of zero-mean, uncorrelated elements with

variance ε^2 $\big($that is, for example: $\Delta Z_{n+1;(i,j)} \sim \mathcal{N}(0, \varepsilon^2)\big)$ for sake of the covariance analysis permits a covariance estimate in which there is uncertainty in the radial direction of the spin-axis direction estimate. However, by construction of the problem, we know that the spin-axis direction in reality has no distribution in the radial direction – it is a unit-norm vector confined to the surface of the unit sphere. To account for this factor, the following approach is adopted. Each covariance estimate $P_{\hat{\boldsymbol{n}}}$ as expressed in the inertial Cartesian frame is transformed into the local East-North-Up (ENU) frame defined by the location on the unit sphere corresponding to the position of the current spin-axis direction estimate $\hat{\boldsymbol{n}}$ (see Fig. 2.2). In this new local ENU frame, the covariance terms involving the "up" direction are set to zero to enforce the unit-norm constraint on the covariance in the radial direction. Finally, the modified covariance matrix expressed in the ENU frame is transformed back to the inertial frame. In particular, the transformation is performed as:

$$P_{\hat{\boldsymbol{n}}}^{ENU} = \left(R^{ENU}\right) P_{\hat{\boldsymbol{n}}} \left(R^{ENU}\right)^T \tag{2.51}$$

where the transformation matrix $R^{ENU} = R^{ENU}\left(\hat{\boldsymbol{n}}\right)$ is given by:

$$\begin{array}{l} X = \hat{\boldsymbol{n}}(1) \\ Y = \hat{\boldsymbol{n}}(2) \\ Z = \hat{\boldsymbol{n}}(3) \end{array} \qquad \longrightarrow \qquad \begin{array}{l} \psi = \tan^{-1}\left(\dfrac{Y}{X}\right) \\ \varphi = \tan^{-1}\left(\dfrac{Z}{\sqrt{X^2+Y^2}}\right) \end{array}$$

$$R^{ENU} = \begin{bmatrix} -\sin(\psi) & \cos(\psi) & 0 \\ -\sin(\varphi)\cos(\psi) & -\sin(\varphi)\sin(\psi) & \cos(\varphi) \\ \cos(\varphi)\cos(\psi) & \cos(\varphi)\sin(\psi) & \sin(\varphi) \end{bmatrix} \tag{2.52}$$

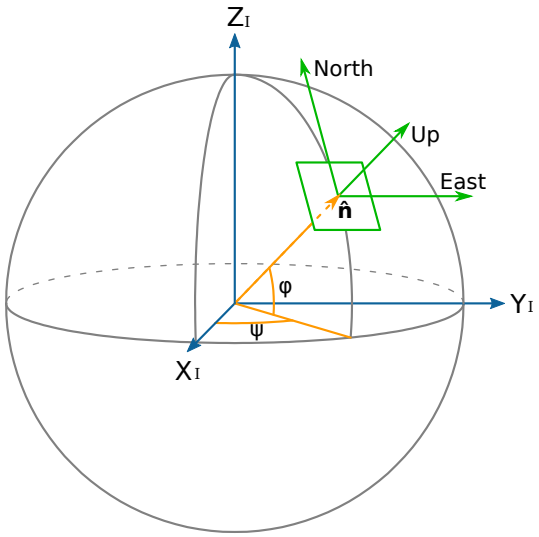


Figure 2.2: Local ENU Frame of $\hat{\boldsymbol{n}}$

Next, the covariance terms involving the "up" direction are set to zero:

$$P_{\hat{\boldsymbol{n}}}^{ENU} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cancelto{0}{p_{1,3}} \\ p_{2,1} & p_{2,2} & \cancelto{0}{p_{2,3}} \\ \cancelto{0}{p_{3,1}} & \cancelto{0}{p_{3,2}} & \cancelto{0}{p_{3,3}} \end{bmatrix} \tag{2.53}$$

$$P_{\hat{\boldsymbol{n}}}^{ENU} = \begin{bmatrix} p_{1,1} & p_{1,2} & 0 \\ p_{2,1} & p_{2,2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.54}$$

Finally, the modified covariance matrix expressed in the local ENU frame $P_{\hat{\boldsymbol{n}}}^{ENU}$ is rotated back to the inertial frame:

$$P_{\hat{\boldsymbol{n}}} = \left(R^{ENU}\right)^T P_{\hat{\boldsymbol{n}}}^{ENU} \left(R^{ENU}\right) \tag{2.55}$$

2.2.4 Variance Estimates for the Angles

Finally, the variances of the three angle estimates $\hat{\theta}$, $\hat{\beta}$, $\hat{\gamma}$ (Eqs. 2.28, 2.29, 2.30, respectively) are derived, under linearization. The Jacobians of $\hat{\beta}$ and $\hat{\gamma}$ with respect to the spin-axis direction $\hat{\boldsymbol{n}}$ are given by:

$$J_{\hat{\beta}} = \frac{\partial \hat{\beta}}{\partial \hat{\boldsymbol{n}}} = \begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{1-\hat{\boldsymbol{n}}(3)^2}} \end{bmatrix} \tag{2.56}$$

$$J_{\hat{\gamma}} = \frac{\partial \hat{\gamma}}{\partial \hat{\boldsymbol{n}}} = \frac{1}{\hat{\boldsymbol{n}}(1)^2 + \hat{\boldsymbol{n}}(2)^2} \begin{bmatrix} -\hat{\boldsymbol{n}}(2) & \hat{\boldsymbol{n}}(1) & 0 \end{bmatrix} \tag{2.57}$$

Therefore, the variance estimates for $\hat{\beta}$ and $\hat{\gamma}$ can be computed from the spin-axis direction variance estimate $P_{\hat{\boldsymbol{n}}}$ by:

$$P_{\hat{\beta}} = \sigma_{\hat{\beta}}^2 = J_{\hat{\beta}} P_{\hat{\boldsymbol{n}}} J_{\hat{\beta}}^T \tag{2.58}$$

$$P_{\hat{\gamma}} = \sigma_{\hat{\gamma}}^2 = J_{\hat{\gamma}} P_{\hat{\boldsymbol{n}}} J_{\hat{\gamma}}^T \tag{2.59}$$

The variance derivation for $\hat{\theta}$ is a little bit more involved, as $\hat{\theta}$ is a function of not just $\hat{\boldsymbol{n}} = \boldsymbol{u}_1 \times \boldsymbol{u}_2$, but also $\boldsymbol{u}_1$ (see Eq. 2.28). First, recall the stacked vector $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1^T & \boldsymbol{u}_2^T \end{bmatrix}^T$ with covariance matrix P (see Eq. 2.45). When constructing P however, both sub-covariance matrices P_1 and P_2 as expressed in the inertial Cartesian frame need to be transformed into local ENU frames defined by the locations on the unit sphere corresponding to the positions of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, respectively. In these new local ENU frames, the covariance

terms involving the "up" directions are set to zero to enforce the unit-norm constraint on the covariances in the respective radial directions. Finally, the modified covariance matrices expressed in the ENU frames are transformed back to the inertial frame. This procedure follows the procedure presented in Sec. 2.2.3 for $\hat{\boldsymbol{n}}$ and $P_{\hat{\boldsymbol{n}}}$.

The Jacobian of $\hat{\theta}$ with respect to $\boldsymbol{u}$ is given by:

$$J_{\hat{\theta}} = \frac{\partial \hat{\theta}}{\partial \boldsymbol{u}} = \frac{2/(n+1)}{\sin^2\left(\hat{\beta}\right)\sqrt{\sin^2\left(\hat{\beta}\right) - \left(\boldsymbol{u}_1(3)\right)^2}} \begin{bmatrix} \boldsymbol{u}_1(3)\boldsymbol{u}_2(2)\hat{\boldsymbol{n}}(3) \\ -\boldsymbol{u}_1(3)\boldsymbol{u}_2(1)\hat{\boldsymbol{n}}(3) \\ \sin^2\left(\hat{\beta}\right) \\ -\boldsymbol{u}_1(2)\boldsymbol{u}_1(3)\hat{\boldsymbol{n}}(3) \\ \boldsymbol{u}_1(1)\boldsymbol{u}_1(3)\hat{\boldsymbol{n}}(3) \\ 0 \end{bmatrix}^T \tag{2.60}$$

where $(n+1)$ is the total number of measurements. Therefore, the variance estimate for $\hat{\theta}$ can be computed by:

$$P_{\hat{\theta}} = \sigma_{\hat{\theta}}^2 = J_{\hat{\theta}} P J_{\hat{\theta}}^T \tag{2.61}$$

2.3 Algorithm Outline

The general algorithm to generate the spin-axis direction estimate and accompanying covariance estimate is summarized in Alg. 1. Details of the numerical checks to resolve quadrant ambiguities and eigenvalue swapping are omitted for conciseness.

Algorithm 1: Outline

Result: Spin-axis direction estimate $\hat{\boldsymbol{n}}$ and accompanying covariance estimate $P_{\hat{\boldsymbol{n}}}$ for $(n+1)$ measurements

initialization ;

while *running* **do**

- obtain new measurement $\boldsymbol{z}_n$;
- stack new measurement $\boldsymbol{z}_n$ into measurement matrix $\boldsymbol{Z}_{n+1}$ (Eq. 2.25) ;
- perform SVD on measurement matrix $\boldsymbol{Z}_{n+1}$ (Eq. 2.40) ;
- check for eigenvalue swapping ;
- estimate spin-axis direction $\hat{\boldsymbol{n}}$ (Eq. 2.27) ;
- extract Euler angle estimates $\hat{\theta}$, $\hat{\beta}$, $\hat{\gamma}$ (Eqs. 2.28, 2.29, 2.30) ;
- resolve quandrant ambiguities ;
- compute standard deviation $\sigma_{3,i}$ of each element of third left singular vector $\boldsymbol{u}_3$ (Eq. 2.42) ;
- compute covariance matrix $P_{\hat{\boldsymbol{n}}}$ of spin-axis direction estimate $\hat{\boldsymbol{n}}$ (Eq. 2.50) ;
- transform covariance matrix $P_{\hat{\boldsymbol{n}}}$ to local ENU frame $P_{\hat{\boldsymbol{n}}}^{ENU}$ (Eq. 2.51) ;
- zero-out covariance terms of $P_{\hat{\boldsymbol{n}}}^{ENU}$ associated with "up" direction (Eq. 2.53) ;
- transform modified covariance matrix $P_{\hat{\boldsymbol{n}}}^{ENU}$ back to inertial frame $P_{\hat{\boldsymbol{n}}}$ (Eq. 2.55) ;
- output spin-axis direction estimate $\hat{\boldsymbol{n}}$ and accompanying covariance estimate $P_{\hat{\boldsymbol{n}}}$;

end

2.4 Results

An exhaustive Monte Carlo (MC) analysis is conducted to numerically evaluate the covariance estimates for varying values of rotation angles θ, β, γ and varying measurement noise gains.

2.4.1 Noise Generation

The noisy measurement $\boldsymbol{z}_k = R_\eta(k)\bar{\boldsymbol{z}}_k$ at each time $t = t_k$ is drawn from the three-dimensional (spherical) von Mises-Fisher probability distribution centered about the noise-free measurement $\bar{\boldsymbol{z}}_k$ with concentration parameter κ:

$$p_{vmf}\left(\boldsymbol{z};\ \bar{\boldsymbol{z}}_k,\ \kappa\right) = C_{vmf}(\kappa)\exp\left(\kappa\bar{\boldsymbol{z}}_k^T\boldsymbol{z}\right) \tag{2.62}$$

where $C_{vmf}(\kappa)$ is given by:

$$C_{vmf}(\kappa) = \frac{\kappa}{4\pi\sinh(\kappa)} \tag{2.63}$$

for the three-dimensional case.

Recall that for the stacked measurement matrix Z_{n+1} in Eq. 2.39, the linear perturbation term ΔZ_{n+1} is assumed to be composed of zero-mean, uncorrelated elements with variance ε^2. The variance ε^2 is set to be related to the measurement noise concentration parameter κ as:

$$\varepsilon^2 = \frac{1}{\kappa} \tag{2.64}$$

That is, the concentration parameter κ in the von Mises-Fisher distribution is inversely proportional to the variance term in a regular Gaussian distribution.

2.4.2 Monte Carlo Analysis

Several Monte Carlo test are performed to validate the performance of the angle estimation and the accuracy of the spin-axis direction covariance estimation. The following parameters are specified for each run:

- Number of Monte Carlo trials: `n_mc`
- In-plane rotation angle between measurements: θ
- Obliquity angle: β
- Precession angle: γ
- von Mises-Fisher noise concentration: $\kappa = \frac{1}{\varepsilon^2}$
- Maximum batch size analyzed: `max_batch_size`

For each test, the angles θ, β, γ are fixed, and the `max_batch_size` value is chosen to be 75% of a 180° in-plane rotation by steps of angular size θ ($0.75 * 180°/\theta$).

Each test presents the following two plots. The first plot is an "Angle Estimates" plot that contains: (1) the true angles θ, β, γ, (2) each Monte Carlo run's angle estimates, (3) the Monte Carlo mean angle estimates, (4) the Monte Carlo mean angle estimates $\pm 3\sigma$ predicted standard deviations, and (5) the Monte Carlo mean angle estimates $\pm 3\sigma$ sample standard deviations. The second plot is a "$\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ Estimate" plot that contains the following for each element of the spin-axis direction estimate $\hat{\boldsymbol{n}}$: (1) the Monte Carlo sample standard deviation, (2) each Monte Carlo run's predicted standard deviation, and (3) the Monte Carlo mean predicted standard deviation. It is important to note that the predicted spin-axis direction estimate's standard deviation (in general, covariance matrix) is dependent on the Monte Carlo realization. The covariance estimate $P_{\hat{\boldsymbol{n}}}$ depends on the specific angle estimates ($\hat{\theta}$, $\hat{\beta}$, $\hat{\gamma}$), which in turn depend on the specific noise realizations in each Monte Carlo run. Therefore, we present both the Monte Carlo standard deviation estimates, and the mean standard deviation estimate, which is averaged over all the Monte Carlo runs.

2.4.2.1 Illustration

An illustrative Monte Carlo simulation for the distributions of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\hat{\boldsymbol{n}}$ is depicted in Fig. 2.3. For this 200-run simulation over 54 time-steps, the underlying Euler angle parameters are set to be: $\theta = 2.5°$, $\beta = 10°$, and $\gamma = 15°$. At each time-step for each of the 200 Monte Carlo runs, an uncorrelated multiplicative noise rotation matrix is generated according to

the spherical ($\mathbb{R}^3$) von Mises-Fisher distribution with concentration parameter $\kappa = 700$ (see Noise Generation section). The blue circles represent the final distribution of $\boldsymbol{u}_1$, the red circles represent the final distribution of $\boldsymbol{u}_2$, and the light green circles represent the final distribution of $\hat{\boldsymbol{n}}$. The blue, red, and light green vectors indicate the mean $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\hat{\boldsymbol{n}}$ vectors, respectively, averaged over all Monte Carlo runs. The smaller circles represent the superimposed noisy measurement vectors $\boldsymbol{z}_k$ over all Monte Carlo runs, where the gradient coloring indicates time from start (teal) to finish (orange).

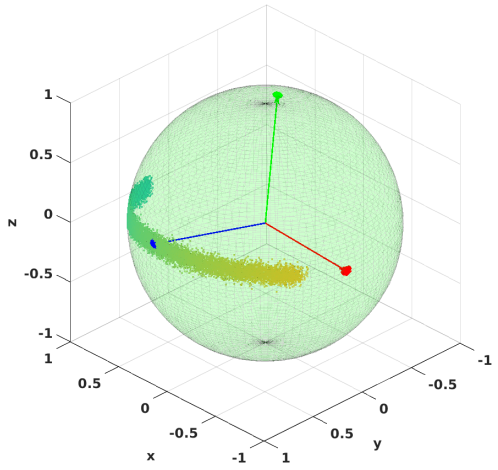


Figure 2.3: Monte Carlo Simulation for $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\hat{\boldsymbol{n}}$

We note that the distributions of the first two left singular vectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ can vary in both the direction of rotation, and normal to the direction of rotation (hence, forming uncertainty "ellipses" on the surface of the unit-sphere). However, uncertainty in the direction of rotation does not affect the

distribution of the spin-axis direction estimate $\hat{\boldsymbol{n}}$ because of the cross-product relationship (Eqs. 2.4, 2.47).

2.4.2.2 Test 1

The first Monte Carlo test is performed with the following parameter values in Table 2.1. Several salient points are observed in the results. In Fig. 2.4a, every Monte Carlo $\hat{\theta}$ estimate converges to the true θ for batch sizes larger than approximately 35. The mean $\hat{\beta}$ converges to the true β by batch size of approximately 20. The mean $\hat{\gamma}$ also converges to the true γ for batch sizes larger than approximately 40. The mean predicted 3σ for $\hat{\theta}$ and $\hat{\beta}$ both converge to the respective sample values by approximately batch size 25. However, for $\hat{\gamma}$, the mean predicted 3σ remains a slightly higher bound to the sample value for most of the simulation, though the mean predicted value does eventually converge to the sample value. In Fig. 2.4b, the mean standard deviation estimate $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ converges to the sample standard deviation for batch sizes larger than about 25. This approximately correlates to the convergence of the mean $\hat{\beta}$ to the true β.

n_mc	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{\kappa} = \frac{1}{\varepsilon^2}$	max_batch_size
200	1°	10°	15°	700	135

Table 2.1: MC Test 1 Parameters

2.4.2.3 Test 2

The second Monte Carlo test is performed with the following parameter values in Table 2.2. The only difference between Test 1 and Test 2 is the decrease of the noise concentration parameter κ from 700 to 350. This corresponds to increasing the noise variance by a factor of two.

The doubling of the noise variance mainly manifests as larger variances in the distributions of the three Euler angles and of $\hat{\boldsymbol{n}}$, and a delayed convergence of the mean predicted 3σ for $\hat{\beta}$ to the sample value (by approximately batch size 30 instead of 25) as seen in Fig. 2.5a. This delayed convergence is also reflected in the later convergence of the mean standard deviation estimate $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ to the sample value by approximately batch size 30, as seen in Fig. 2.5b.

n_mc	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{\kappa} = \frac{\mathbf{1}}{\boldsymbol{\varepsilon}^{\mathbf{2}}}$	max_batch_size
200	1°	10°	15°	350	135

Table 2.2: MC Test 2 Parameters

2.4.2.4 Test 3

The third Monte Carlo test is performed with the following parameter values in Table 2.3. This Test 3 differs from Test 1 in two ways. First, the in-plane rotation between measurements θ is halved from 1° to 0.5°. Second, the maximum batch size is accordingly doubled from 135 to 270. Thus, the maximum angular displacement in the plane-of-rotation is kept the same (135°).

As seen in Figs. 2.6a and 2.6b, the main difference decreasing the spin rate θ causes is a later convergence (in terms of number of measurements) of the angle estimates to the respective truths, and correspondingly a later convergence of the mean predicted Euler angle 3σ values and the right-hand-plot standard deviation estimates to the respective sample values. We can draw the following conclusion: the magnitude of the true in-plane rotation rate θ does not matter much as long as sufficient angular displacement in the plane-of-rotation is observed.

n_mc	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{\kappa} = \frac{1}{\varepsilon^2}$	max_batch_size
200	0.5°	10°	15°	700	270

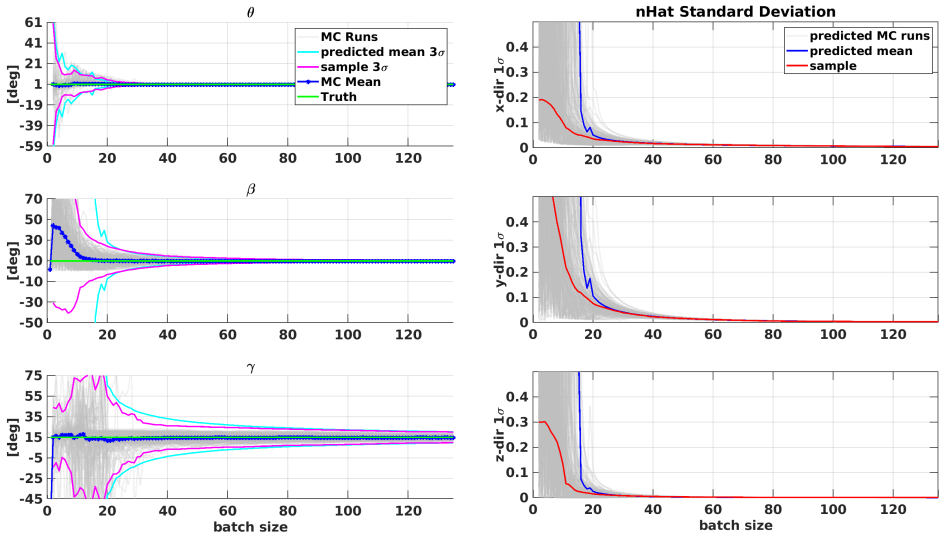
Table 2.3: MC Test 3 Parameters

2.4.2.5 Test 4

The fourth Monte Carlo test is performed with the following parameter values in Table 2.4. The only difference between Test 3 and Test 4 is the decrease of the noise concentration parameter κ from 700 to 350. This corresponds to increasing the noise variance by a factor of two. We note that for this slower rotation angle $\theta = 0.5°$ between measurements, the effect of doubling the noise variance only slightly delays at which batch size do the mean predicted Euler angle 3σ values and the right-hand-plot mean standard deviation estimates converge to the respective sample values.

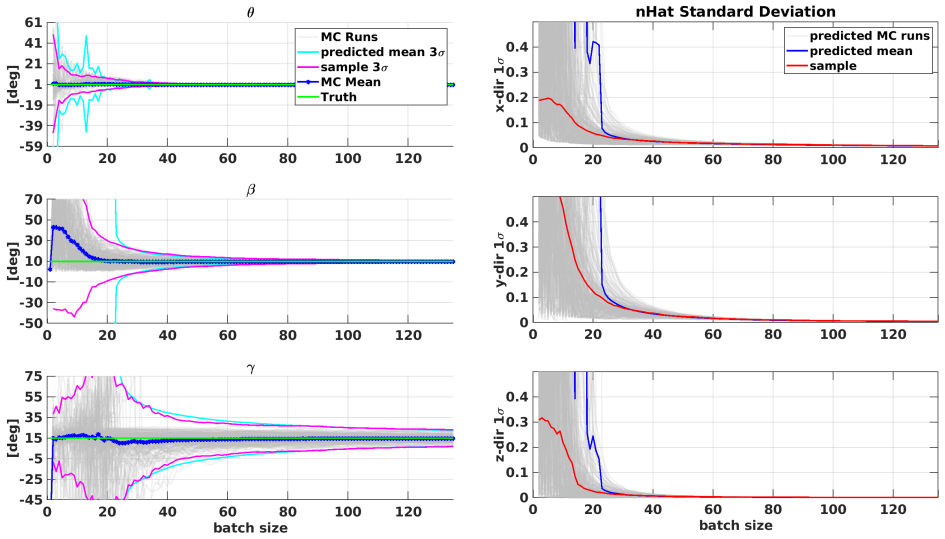
n_mc	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{\kappa} = \frac{1}{\varepsilon^2}$	max_batch_size
200	0.5°	10°	15°	350	270

Table 2.4: MC Test 4 Parameters



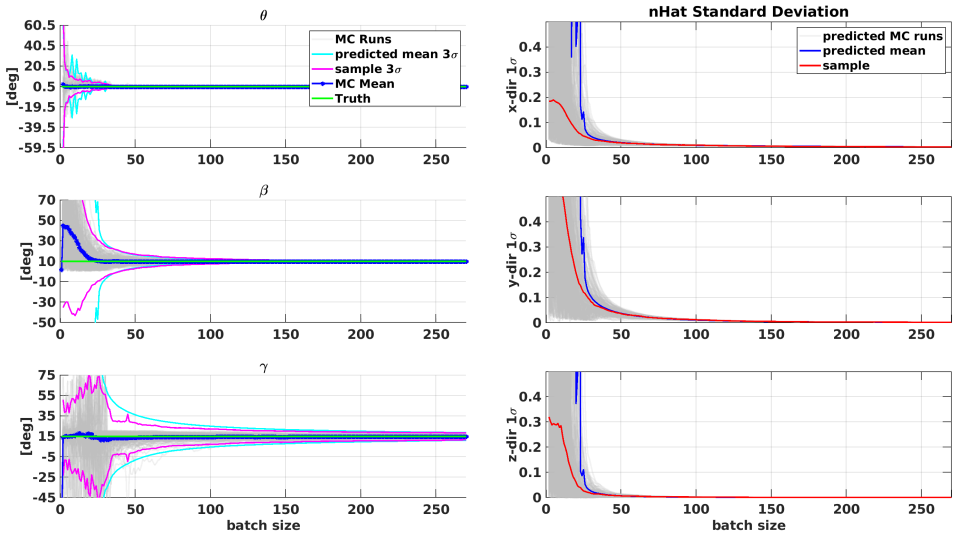
(a) Angle Estimates (b) $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ Estimate

Figure 2.4: Test 1



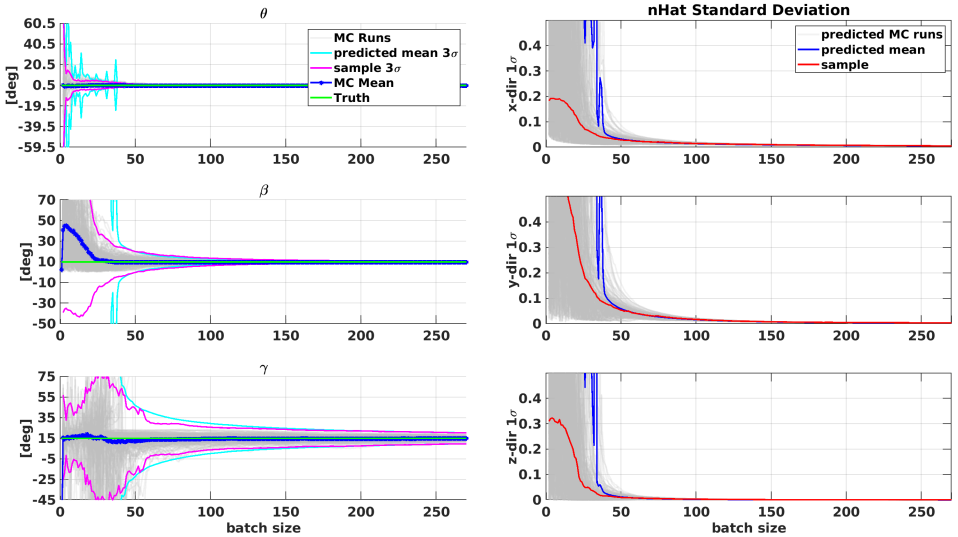
(a) Angle Estimates (b) $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ Estimate

Figure 2.5: Test 2



(a) Angle Estimates (b) $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ Estimate

Figure 2.6: Test 3



(a) Angle Estimates (b) $\boldsymbol{\sigma}_{\hat{\boldsymbol{n}}}$ Estimate

Figure 2.7: Test 4

2.5 Conclusion

This chapter derives a batch estimator that ingests a time-series of relative orientation measurements at fixed time-steps to estimate not only the angular velocity magnitude and spin-axis direction of a target body undergoing constant-rate pure-spin, but also the accompanying uncertainty bounds for the resulting spin-axis direction estimate under reasonable assumptions. In addition, variance estimates for the three Euler angles parameterizing the angular velocity vector are provided. Instead of a recursive filtering methodology, this work has pursued a batch process that is well-suited to exploit the geometric properties intrinsic to this problem via the singular value decomposition (SVD). This batch approach relinquishes the need for an iterative scheme to compute the error bounds upon the estimated spin-axis direction. Importantly, this work has introduced explicit analytical expressions for the singular values and left singular vectors that result from SVD analysis for the noise-free case representing ideal measurements. This work has built upon the recent literature by providing an explicit, non-recursive covariance bound for the estimated spin-axis direction by itself that does not need iteration. Monte Carlo simulations have demonstrated the in-plane rotation rate and spin-axis direction estimates to be zero-mean, and the spin-axis direction covariance estimate to match the sample statistics, for sufficiently large batch sizes.

Some fruitful directions for further work include: (1) accounting for a varying spin-axis direction and in-plane rotation rate, and (2) accounting for non-constant measurement covariance over multiple measurements.

References

[1] B. Banker and S. Askew, "Seeker 1.0: Prototype robotic free flying inspector mission overview," 2019.

[2] N. Dhamani, G. Martin, C. Schubert, P. Singh, N. Hatten, and M. R. Akella, "Applications of machine learning and monocular vision for autonomous on-orbit proximity operations," in *AIAA Scitech 2020 Forum*, p. 1376, 2020.

[3] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[4] K. Black, S. Shankar, D. Fonseka, J. Deutsch, A. Dhir, and M. R. Akella, "Real-time, flight-ready, non-cooperative spacecraft pose estimation using monocular imagery," *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021 (Accepted For Publication).

[5] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.

[6] S. Agarwal, K. Mierle, and Others, "Ceres solver." `http://ceres-solver.org`.

[7] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[10] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of guidance, control, and dynamics*, vol. 26, no. 2, pp. 311–317, 2003.

[11] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[12] E. Leffens, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.

[13] M. Almeida, D. Mortari, R. Zanetti, and M. Akella, "Quatera: The quaternion regression algorithm," in *Journal of Guidance,Control, and Dynamics*, 2020. Accepted, In Press.

[14] S. Kaki and M. R. Akella, "Uncertainty quantification for a batch filter: Estimating the angular velocity of a rigid-body in near pure-spin condi-

tion," *2020 AAS/AIAA Astrodynamics Specialist Conference*, vol. AAS 20-652, 2020.

[15] E. Gai, K. Daly, J. Harrison, and L. Lemos, "Star-sensor-based satellite attitude/attitude rate estimator," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 560–565, 1985.

[16] D. Mortari and M. Akella, "Discrete and continuous time adaptive angular velocity estimators," in *Proceedings of the AAS/AIAA Space Flight Mechanics Conference, Williamsburg, VA*, 2015.

[17] A. Melman, "Spectral functions for real symmetric toeplitz matrices," *Journal of computational and applied mathematics*, vol. 98, no. 2, pp. 233–243, 1998.

[18] H. Schaub, J. L. Junkins, *et al.*, "Stereographic orientation parameters for attitude dynamics: A generalization of the rodrigues parameters," *Journal of the Astronautical Sciences*, vol. 44, no. 1, pp. 1–19, 1996.

[19] B. P. Epps and E. M. Krivitzky, "Singular value decomposition of noisy data: mode corruption," *Experiments in Fluids*, vol. 60, no. 8, p. 121, 2019.